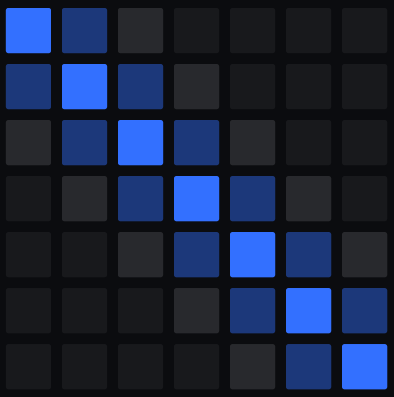
Parameter

Sample report

Penetration test report.

Prepared for Acme Corporation



[Assessment]

Black Box Web Application Penetration Test

[Target]

app.acme.example (+1 more)

[Issued]

March 6, 2026

[Confidential]

00

Contents.

01

Executive summary.

Parameter confirmed 7 issues against app.acme.example during the assessment window (March 2, 2026 - March 6, 2026). Of those, 4 are critical or high severity and are ranked below for immediate attention.

RISK POSTURE

2 resolved

7

findings

1 Critical 3 High 2 Medium 1 Low

ASSESSMENT

Target	app.acme.example
Method	Black Box Web Application Penetration Test
Window	March 2, 2026 - March 6, 2026

PRIORITY ACTIONS

SQL injection in the account lookup endpoint — Critical — ACME-001

The account lookup endpoint interpolates the submitted email address directly into its SQL statement.

Insecure direct object reference on invoice downloads — High — ACME-003

The invoice download endpoint authenticates the caller but never checks that the invoice belongs to their organisation.

Server-side request forgery in the webhook validator — High — ACME-004

Before saving a webhook, the application fetches the supplied URL to confirm it responds.

02

Methodology.

How the assessment was run: the frameworks it draws on, the surface it covered, and where human verification sits in the loop.

TESTING APPROACH

This assessment was conducted using Parameter's proprietary penetration testing framework, which is built on and informed by accredited, industry-recognized security standards and refined through real-world exploitation research.

The framework draws from attested frameworks such as OWASP Testing Guide, OWASP Top 10, NIST risk-based security principles, and MITRE ATT&CK where applicable.

These standards serve as reference baselines, not rigid checklists. Parameter extends them with internal testing frameworks, attack heuristics, and validation criteria developed from production security assessments.

Execution is performed using a combination of AI-driven analysis and exploitation and human-led verification. Parameter leverages custom-trained large language models (LLMs) and AI agents to analyze the codebase, reason about control flow and trust boundaries, and generate exploit hypotheses aligned to the above frameworks. Human security engineers retain accountability for validation, execution, and impact assessment of all findings.

AREAS OF TESTING

The following security domains were evaluated during this assessment:

- Authentication and Session Management
- Authorization and Access Control
- Input Validation and Output Encoding
- Cryptographic Implementation
- Error Handling and Logging
- Business Logic Vulnerabilities
- API Security
- Configuration and Deployment

EXECUTION PROCESS

1. Attack surface mapping using proprietary tooling

2. AI-assisted analysis using custom-trained LLMs and agents to identify risk patterns, trust boundaries, and exploit hypotheses
3. AI-assisted and manual exploit development and execution to validate real-world impact
4. Human verification and impact assessment for all successful exploits
5. Severity scoring using CVSS v3.1
6. Remediation verification through targeted retesting and code review

AI ASSURANCE STATEMENT

Testing followed a structured and repeatable process defined by Parameter's internal methodology:

All testing was conducted within the authorized scope and followed responsible security testing practices. This assessment represents a point-in-time evaluation of the application during the testing window and does not guarantee the absence of all security vulnerabilities.

Parameter uses custom-trained large language models (LLMs) and AI agents to augment its proprietary penetration testing methodology. AI is used to assist with codebase analysis, vulnerability hypothesis generation, and exploit development, operating within predefined guardrails informed by accredited security frameworks.

All AI-assisted outputs are treated as inputs only. Human security engineers execute, validate, and assess all exploitation activities, confirm impact, eliminate false positives, and determine severity and remediation guidance. No findings are reported without successful exploitation or direct manual confirmation.

AI systems operate strictly within the authorized testing scope and are subject to access controls and oversight. While AI improves coverage and efficiency, final accountability for findings and conclusions rests with Parameter's security research team, and this assessment remains a point-in-time evaluation.

ATTACK SURFACE OVERVIEW

Target Type	Target Details
Web Application	https://app.acme.example
Web Application	https://api.acme.example

03

Findings.

Every issue confirmed during the engagement, ordered by severity. Full technical detail follows in the next section.

ID	FINDING	SEVERITY	STATUS
ACME-001	SQL injection in the account lookup endpoint	Critical	Open
ACME-002	Stored cross-site scripting in saved search names	High	Resolved
ACME-003	Insecure direct object reference on invoice downloads	High	Open
ACME-004	Server-side request forgery in the webhook validator	High	Open
ACME-005	No rate limiting on the authentication endpoint	Medium	Open
ACME-006	Session cookie issued without the Secure attribute	Medium	Resolved
ACME-007	Stack traces disclosed in API error responses	Low	Open

HOW SEVERITY IS RATED

Critical	Direct compromise of data, accounts or infrastructure with no meaningful precondition.	immediate
High	Significant compromise requiring a common precondition such as an authenticated session.	7 days
Medium	Meaningful weakening of a control, or a step an attacker chains toward a larger goal.	30 days
Low	Limited impact in isolation; worth fixing as hygiene.	90 days

04

Detailed findings.

Description, reproduction, impact and remediation for every confirmed issue, grouped by severity.

CRITICAL & HIGH SEVERITY

Critical • ACME-001

SQL injection in the account lookup endpoint

STATUS	CWE	OWASP	IDENTIFIED
Open	CWE-89	A03:2021 - Injection	March 2, 2026

The account lookup endpoint interpolates the submitted email address directly into its SQL statement. Any input is parsed as query syntax, so an unauthenticated attacker controls the `WHERE` clause and can return arbitrary rows — or terminate the statement and run a second one.

Reproduction

1. Send a lookup request with `' OR '1'='1` as the email address.
2. Observe that the first row in the table is returned rather than a not-found response.
3. Replace the payload with `'; SELECT pg_sleep(5); --` to confirm that a second statement executes.

```
POST /api/accounts/lookup HTTP/1.1
Host: api.acme.example
Content-Type: application/json

{"email": "' OR '1'='1"}
```

Impact

Every account record is readable without credentials, including email addresses and password hashes. The application connects as a role that can also write, so an attacker can create an account, elevate a role, or drop the audit tables that would record the intrusion.

Remediation

Use a parameterised statement so user input is never parsed as SQL, disable multi-statement execution on the connection, and move the application to a least-privilege database role.

```
const [account] = await db
  .select()
  .from(accounts)
  .where(eq(accounts.email, input.email))
  .limit(1);
```

High • ACME-002

Stored cross-site scripting in saved search names

STATUS	CWE	OWASP	IDENTIFIED
Resolved	CWE-79	A03:2021 - Injection	March 3, 2026

Saved search names are rendered into the sidebar without contextual encoding. A name containing markup is parsed as HTML, so script stored by one user executes in the session of anyone the search is shared with.

Because saved searches can be shared team-wide, a single crafted name reaches every member of the workspace on their next visit — including administrators, whose session can issue API keys.

Remediation

Render the name as text rather than markup. Where highlighting is required, build the `<mark>` elements as nodes instead of interpolating a string, and add `Content-Security-Policy: script-src 'self'` so an injected inline script is blocked even if encoding is missed.

Verified fixed during the engagement — a retest on 5 March confirmed the payload no longer executes.

High • ACME-003

Insecure direct object reference on invoice downloads

STATUS	CWE	OWASP	IDENTIFIED
Open	CWE-639, CWE-200	A01:2021 - Broken Access Control	March 3, 2026

The invoice download endpoint authenticates the caller but never checks that the invoice belongs to their organisation. Any signed-in user can retrieve another organisation's invoice by changing the identifier in the path.

Identifiers are sequential, so the full invoice history across every tenant is enumerable from a single trial account.

```
GET /api/invoices/48213/pdf HTTP/1.1
Cookie: sid=<any valid session>
```

Impact

Cross-tenant disclosure of billing records: legal entity names, addresses, line items and contract values. For a multi-tenant product this is both a confidentiality breach and a contractual exposure.

Remediation

Scope the query by the caller's organisation, and issue opaque identifiers so enumeration is not possible even if an authorisation check is missed later.

High • ACME-004

Server-side request forgery in the webhook validator

STATUS	CWE	OWASP	IDENTIFIED
Open	CWE-918	A10:2021 - Server-Side Request Forgery	March 4, 2026

Before saving a webhook, the application fetches the supplied URL to confirm it responds. The URL is not validated against an allow-list and redirects are followed, so an attacker can reach internal services from the application's network position.

The cloud metadata endpoint was reachable during testing, returning instance identity information.

Reproduction

1. Register a webhook pointing at `http://169.254.169.254/latest/meta-data/`.
2. Trigger validation and read the response surfaced in the UI error message.

Remediation

Resolve the hostname and reject any address in a private, loopback, link-local or metadata range before connecting; re-check after every redirect; and make the validator's response opaque to the caller so it cannot be used as a read primitive.

MEDIUM SEVERITY

Medium • ACME-005

No rate limiting on the authentication endpoint

STATUS	CWE	OWASP	IDENTIFIED
Open	CWE-307	A07:2021 - Identification and Authentication Failures	March 4, 2026

The login endpoint accepts unlimited attempts per account and per source address. Credential stuffing against a breached-password list is bounded only by network throughput.

ATTEMPTS	WINDOW	RESULT
500	60s	All processed, no throttling
5,000	10m	All processed, no lockout

No lockout, no challenge escalation, and no alert reached the security team during testing.

Remediation

Apply a sliding-window limit keyed on both the account and the source address, and escalate to a challenge before locking out — a lockout keyed only on the account is itself a denial-of-service vector.

Medium • ACME-006

Session cookie issued without the Secure attribute

STATUS	CWE	OWASP	IDENTIFIED
Resolved	CWE-614, CWE-319	A05:2021 - Security Misconfiguration	March 5, 2026

The session cookie is set with `HttpOnly` but without `Secure`, so it is transmitted over plaintext HTTP if a request ever reaches the origin without TLS. HSTS is not sent, so a first visit typed without a scheme is exposed.

```
Set-Cookie: sid=...; Path=/; HttpOnly; SameSite=Lax
```

Remediation

Add `Secure` to the cookie and send `Strict-Transport-Security: max-age=31536000; includeSubDomains` so the browser never attempts a plaintext request.

Verified fixed during the engagement — a retest on 6 March confirmed both headers.

LOW & INFORMATIONAL

Low • ACME-007

Stack traces disclosed in API error responses

STATUS	CWE	OWASP	IDENTIFIED
Open	CWE-209	A05:2021 - Security Misconfiguration	March 5, 2026

Unhandled exceptions return the full stack trace to the client, including absolute file paths, framework versions and the internal address of the database.

```
{"error":"connect ECONNREFUSED 10.4.2.19:5432","stack":"at Connection._
handleErrorWhileConnecting (/srv/app/node_modules/pg/lib/client.js:327:19)"}
```

This grants no access on its own, but it shortens reconnaissance: it confirms the datastore, its private address, and the exact dependency versions to check against public advisories.

Remediation

Return a generic message and a correlation identifier to the client; log the full detail server-side only.

05

Conclusion.

VERDICT

Critical issues require immediate attention

1 critical severity vulnerability remains open. Immediate remediation is strongly recommended to reduce risk exposure.

This security assessment identified a total of 7 vulnerabilities across all severity levels. Of these, 2 have been remediated and 5 remain open.

REMEDIATED DURING THE ENGAGEMENT

The following vulnerabilities have been successfully remediated during or after this assessment:

- Stored cross-site scripting in saved search names (ACME-002)
- Session cookie issued without the Secure attribute (ACME-006)

06

Scope.

ABOUT THIS DOCUMENT

Sample report. Acme Corporation is a fictional client and every finding, target and date in this document is illustrative. It shows the structure and depth of a Parameter assessment; it is not a record of a real engagement.

TARGETS

Web application — https://app.acme.example

Web application — https://api.acme.example

Testing was limited to the targets listed above. No representation is made about systems, components or functionality outside this scope.

POINT IN TIME

This assessment reflects the security posture of app.acme.example between March 2, 2026 - March 6, 2026. Changes to the application, its infrastructure or the wider threat landscape after this date are not reflected here.

LIMITATIONS

An assessment does not guarantee the absence of all vulnerabilities. Undiscovered issues may remain despite reasonable testing effort. Findings and recommendations are based on technologies and threats known at the time of testing.

CONFIDENTIALITY

This document is confidential and intended solely for Acme Corporation. Distribution to any other party, or to third parties to whom Acme Corporation has given explicit written permission, is prohibited. Anytool, Inc. assumes no liability for unauthorised use or misuse of this document. If you received this document in error, close it and notify Anytool, Inc.

DOCUMENT CONTROL PAR-2026-ACM-0001

Classification	Confidential
Assessment	Black Box Web Application Penetration Test
Prepared by	Parameter — automated assessment, human verified
Distribution	Acme Corporation
Issued	March 6, 2026

Glossary.

This glossary provides definitions for technical terms used throughout this report.

Term	Definition
CVSS	Common Vulnerability Scoring System - An open framework for communicating the characteristics and severity of software vulnerabilities.
CWE	Common Weakness Enumeration - A community-developed list of software and hardware weakness types.
OWASP	Open Web Application Security Project - A nonprofit foundation that works to improve the security of software.
XSS	Cross-Site Scripting - A type of injection attack where malicious scripts are injected into trusted websites.
SQLi	SQL Injection - A code injection technique used to attack data-driven applications by inserting malicious SQL statements.
CSRF	Cross-Site Request Forgery - An attack that forces authenticated users to submit requests they did not intend to make.
IDOR	Insecure Direct Object Reference - A vulnerability where user-controlled input is used to access objects directly.
RCE	Remote Code Execution - A vulnerability that allows an attacker to execute arbitrary code on a target system.
SSRF	Server-Side Request Forgery - A vulnerability where an attacker can induce the server to make requests to unintended locations.
Black Box Testing	Security testing performed without knowledge of internal implementation details.
White Box Testing	Security testing performed with full access to source code and internal documentation.
Penetration Test	An authorized simulated cyberattack on a computer system, performed to evaluate system security.

Parameter

security@parameter.ai

Confidential report prepared for Acme Corporation.